Analysis of Neocognitron of Neural Network Method in the String Recognition

Amit Kumar Gupta [1], Yash Pal Singh [2]
[1] KIET/ MCA Department, Ghaziabad (U.P.), India
Email: amitid29@gmail.com
[2] BIET / CSE Department, Bundelkhand, Jhansi (U.P.), India
Email: yash_biet@yahoo.co.in

Abstract: **This paper aims that analysing neural network method in pattern recognition. A neural network is a processing device, whose design was inspired by the design and functioning of human brain and their components. The proposed solutions focus on applying Neocognitron Algorithm model for pattern recognition. The primary function of which is to retrieve in a pattern stored in memory, when an incomplete or noisy version of that pattern is presented. An associative memory is a storehouse of associated patterns that are encoded in some form. In auto-association, an input pattern is associated with itself and the states of input and output units coincide. When the storehouse is incited with a given distorted or partial pattern, the associated pattern pair stored in its perfect form is recalled. Pattern recognition techniques are associated a symbolic identity with the image of the pattern. This problem of replication of patterns by machines (computers) involves the machine printed patterns. There is no idle memory containing data and programmed, but each neuron is programmed and continuously active.**

Keywords: Neural network, machine printed string, pattern recognition, A perceptron-type network, Learning, Recognition, Connection.

I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the biological nervous systems, such as the brain. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. A Neural Network is configured for pattern recognition or data classification, through a learning process. In biological systems, Learning involves adjustments to the synaptic connections that exist between the neurons. Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements working in parallel to solve a specific problem. Neural networks learn by example. A neuron has many inputs and one output. The neuron has two modes of operation (i) the training mode and (ii) the using mode. In the training mode, the neuron can be trained for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the training rule is used. Neural network has many applications. The most likely applications for the neural networks are (1) Classification (2) Association and (3) Reasoning. An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. Four significant approaches to PR have evolved. These are [5].

Statistical pattern recognition:

Here, the problem is posed as one of composite hypothesis testing, each hypothesis pertaining to the premise, of the datum having originated from a particular class or as one of regression from the space of measurements to the space of classes. The statistical methods for solving the same involve the computation other class conditional probability densities, which remains the main hurdle in this approach. The statistical approach is one of the oldest, and still widely used [8].

Syntactic pattern recognition:

In syntactic pattern recognition, each pattern is assumed to be composed of sub-pattern or primitives string together in accordance with the generation rules of a grammar string of the associated class. Class identifications accomplished by way of parsing operations using automata corresponding to the various grammars [15, 16]. Parser design and grammatical inference are two difficult issues associated with this approach to PR and are responsible for its somewhat limited applicability.

Knowledge-based pattern recognition:

This approach to PR [17] is evolved from advances in rule-based system in artificial intelligence (AI). Each rule is in form of a clause that reflects evidence about the presence of a particular class. The sub-problems spawned by the methodology are:

1. How the rule-based may be constructed, and
2. What mechanism might be used to integrate the evidence yielded by the invoked rules?

Neural Pattern Recognition:

Artificial Neural Network (ANN) provides an emerging paradigm in pattern recognition. The field of ANN encompasses a large variety of models [18], all of which have two important string.

1. They are composed of a large number of structurally and functionally similar units called neurons usually connected various configurations by weighted links.
2. The Ann's model parameters are derived from supplied I/O

DOI: 01.IJNS.02.03.200

paired data sets by an estimation process called training.

II. METHODOLOGY

Different neural network algorithms are used for recognizing the pattern. Various algorithms differ in their learning mechanism. Information is stored in the weight matrix of a neural network. Learning is the determination of the weights. All learning methods used for adaptive neural networks can be classified into two major categories: supervised learning and unsupervised learning. Supervised learning incorporates an external teacher. After training the network, we should get the response equal to target response. During the learning process, global information may be required. The aim is to determine a set of weights, which minimizes the error. Unsupervised learning uses no external teacher and is based on clustering of input data. There is no prior information about input's membership in a particular class. The string of the patterns and a history of training are used to assist the network in defining classes. It self-organizes data presented to the network and detects their emergent collective properties. The characteristics of the neurons and initial weights are specified based upon the training method of the network. The pattern sets is applied to the network during the training. The pattern to be recognized are in the form of vector whose elements is obtained from a pattern grid. The elements are either 0 and 1 or -1 and 1. In some of the algorithms, weights are calculated from the pattern presented to the network and in some algorithms, weights are initialized. The network acquires the knowledge from the environment. The network stores the patterns presented during the training in another way it extracts the features of pattern. As a result of this, the information can be retrieved later.

III. PROBLEM STATEMENT

The aim of the paper is that neural network has demonstrated its capability for solving complex pattern recognition problems. Commonly solved problems of pattern have limited scope. Single neural network architecture can recognize only few patterns. In this paper discusses on neural network algorithm with their implementation details for solving pattern recognition problems. The relative performance evaluation of this algorithms has been carried out. The comparisons of algorithm have been performed based on following criteria:

(1) Noise in weights
(2) Noise in inputs
(3) Loss of connections
(4) Missing information and adding information.

IV. NEOCOGNITRON

The Neocognitron is self organized by unsupervised learning and acquires the ability for correct pattern recognition. For self organization of network, the patterns are presented repeatedly to the network. It is not needed to give prior information about the categories to which these patterns should be classified. The Neocognitron itself acquires the ability to classify and correctly recognize these patterns, according to the differences in their shapes. The network has a multilayer structure. Each layer has two planes. One called s-plane, consists of s units. Other the c-plane, consists of c units. The units can have both excitatory and inhibitory connections. The network has forward connections from input layers to output layer and backward connections from the output layer to the input layer. The forward signals are for pattern classification and recognition. The backward signals are for selective attention, pattern segmentation and associated recall [23].

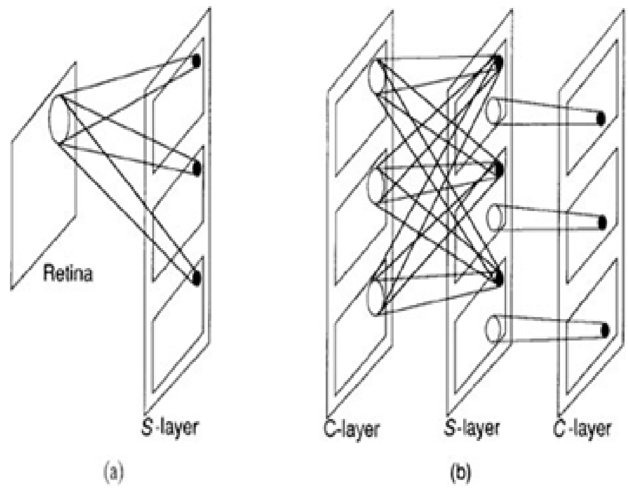

Figure 4.1 shows the character identification process schematically.

Figure 4.1 : This diagram is a schematic representation of the interconnection strategy of the neocognitron. (a) On the first level, each S unit receives input connections from a small region of the retina. Units in corresponding positions on all planes receive input connections from the same region of the retina. The region from which an S-cell receives input connections defines the receptive field of the cell, (b) On intermediate levels, each unit on an s-plane receives input connections from corresponding locations on all C-planes in the previous level. C-celIs have connections from a region of S-cells on the S level preceding it. If the number of C-planes is the same as that of S-planes at that level, then each C-cell has connections from S-cells on a single s-plane. If there are fewer C-planes than S-planes, some C-cells may receive connections from more than one S-plane.

V. NEOCOGNITRON ALGORITHM

Let u(1), u(2), …………u(N) be the excitatory inputs and v be the inhibitory input. The output of the s-cell is given by

$$w = \varphi\left[\frac{1+\sum_{v=1}^{N} a(v)u(v)}{1+bv} - 1\right]$$

Where a(v) and b represent the excitatory and inhibitory interconnecting coefficients respectively. The function $\varphi[x]$ is defined by

DOI: 01.IJNS.02.03. 200

$$\varphi[x] = x \qquad x \geq 0$$

$$x < 0$$

Let e be the sum of all excitatory inputs weighted with interconnecting coefficients and h be the inhibitory input multiplied by the interconnecting coefficients.

$$e = \sum_{v=1}^{N} a(v)u(v)$$

$$h = b.v$$

So w can be written as,

$$w = \varphi\left[\frac{1+e}{1+h} - 1\right] = \varphi\left[\frac{e-h}{1+h}\right]$$

When h<<1, $w = \varphi(e-h)$

In Neocognitron, the interconnecting coefficients a(v) and b increases as learning progresses. If the interconnecting coefficient increases and if e>>1 and h>>1,

$$w = \varphi\left[\frac{e}{h} - 1\right]$$

Where the output depends on the ratio e/h, not on e and h. Therefore, if both the excitatory and inhibitory coefficients increase at the same rate then the output of the cell reaches a certain value. Similarly the input and output characteristics of c-cell is given by,

$$\psi[x] = \frac{x}{\alpha + x} \qquad x \geq 0$$

$$x < 0$$

α is a positive constant which determines the degree saturation of the output. The output of the excitatory cells s and c are connected only to the excitatory input terminals of the other cells. V_s and V_c cells are inhibitory cells, whose output is connected only to the inhibitory input terminals of other cells. A V_s –cell gives an output, which is proportional to the weighted arithmetic mean of all its excitatory inputs. A V_c –cell also has only inhibitory inputs but its output is proportional to the weighted root mean square of its inputs. Let u(1), u(2), ……u(N) be the excitatory inputs and c(1), c(2), ……c(n) be the interconnecting coefficients of its input terminals. The output w of this V_c –cell is defined by

$$w = \sqrt{\sum_{v=1}^{N} c(v)u^2(v)}$$

Storage Capacity. By increasing number of planes in each layer the capacity of the network can be increased. But by increasing no of planes in each layer, the computer can not simulate because of the lack of memory capacity.

VI. Result

Neocognitron algorithm A nine layered network U_0, U_{s1}, U_{c1}, U_{s2}, U_{c2}, U_{s3}, U_{c3}, U_{s4}, and Uc4 is taken. So the network has 4 stages proceeded by an input layer. There are 21 cell planes in each layer U_{s1}-U_{s4}, and 20 cell planes in U_{c1}-U_{c4}, The parameter are listed in table 3.9. As can be seen from the table total number of c-cells in layer U_{c4} is 20 because each c-plane has only one cell. The number of cells in a connectable area is always 5×5 for every s-layer. Hence number of excitatory input to each s-cell is 5×5in layer Us1, and 5×5×20 in layers Us2, Us3 and Us4 because these layers are preceded by c-layers consisting of 20 cell planes in each. During learning five training patterns 0,1,2,3 and 4were presented repeatedly to the input layer. After repeated presentation of these five patterns the Neocognitron acquired the ability to classify these patterns according to the difference in their shape. Each pattern is presented five times. It has been observed that a different cell responds to each pattern.

Table 6.1 shows the Architecture and values of parameters used in Neocognitron.

Layer	Number of Excitatory Cells	Number of Excitatory input Interconnections per cell	Size of an s-column (No of cells per s-column)	r_l	q_l
U_0	19×19	-	-	-	-
U_{s1}	19×19×21	5×5	5×5×20	4.0	1.0
U_{c1}	15×15×20	5×5			
U_{s2}	13×13×21	5×5×20	5×5×20	1.5	12.0
U_{c2}	11×11×20	5×5	-	-	-
U_{s3}	9×9×21	5×5×20	5×5×20	1.5	12.0
U_{c3}	7×7×20	5×5			
U_{s4}	3×3×21	5×5×20	3×3×20	1.5	12.0
U_{c4}	20	3 ×3	-	-	-

The various neural network algorithms differ in their learning mechanism. Some networks learn by supervised learning and in some learning are unsupervised. The capacity of storing patterns and correctly recognizing them differs for different networks. Although if some of the networks are capable of storing same no of patterns, they differ in complexity of their architecture. Total no of neurons needed to classify or recognize the patterns are different for various algorithms. Total no of unknowns to also different for various algorithms. The networks are compared on the basis of these. The performance of various algorithms under different criteria is presented. These criteria are:

A. Noise in Weight

Noise has been introduced in two ways,

1- By adding random numbers to the weight

2- By adding a constant number to all weights of the network

Noise in input: The network is trained with characters without noise. Then by presenting each of the character, the network is tested. It has been observed that the no of characters recognized correctly differs for various algorithms. Noise is introduced to the input vectors at the time of testing and its effect has been observed on various algorithms. This has been done by adding random numbers to the test vector.

DOI: 01.IJNS.02.03.200

B. Loss of Connection

In the network, neurons are interconnected and every interconnection has some interconnecting coefficient called weight. If some these weights are equal to zero then how it is going to effect the classification or recognition, is studied under this section. The number of connections that can be removed such that the network performance is not affected has also been found out for each algorithm. If connection of input neuron's to all the output neuron is removed, and the pixel corresponding to that neuron number is off than it makes no difference. But if that pixel is on, in the output that becomes off. There are two sets of weights. First connecting input neurons to hidden neurons to out put neurons. If the weights connecting 49 input neurons to 15 hidden neurons and weights connecting 15 hidden neurons to 2 output neurons is equated to zero, then for 25 characters actual out is equal to the target output.

C. *Missing Information*

Missing information means some of the on pixels in pattern grid are made off. For the algorithm, how many information we can miss so that the strings can be recognized correctly varies from string to string. We cannot switch off pixel from any place. Which pixel is being switched also matters. For few strings Table-6.2 shows the number of pixels that can be switched off for all the stored strings in algorithm.

TABLE 6.2: MISSING INFORMATION: NO OF PIXELS THAT CAN BE MADE OFF IN THIS ALGORITHM

Character	Neocognitron Algorithm
A	6
I	5
L	4
M	4
P	2
X	5

D. *Adding Information*

There are three weight matrices. Weight connecting inputs to the input layer, input layer to hidden layer, and hidden layer to output layer. If we add 0.0002 to all the weights then for all the 26 characters we get output equal to the target output. If this number is increased then for some of the characters we do not get output same as target output. Table-6.3 shows detailed description about the number of pixels that can be made on for all the strings that can be stored in networks.

TABLE 6.3: ADDING INFORMATION: NO OF PIXELS THAT CAN BE MADE IN THIS ALGORITHM

Character	Neocognitron Algorithm
A	6
I	11
L	5
M	11
P	10
X	9

Demerits

This Neocognitron method , The Network architecture is very complex. It involves a large no of neurons

CONCLUSION

The performance of Quick Prop algorithm has been studied under nine criteria. It has been observed that a certain algorithm performs best under a particular criterion. The algorithm have also been compared based on the number of neurons and the number of unknowns to be computed. The detailed description in Table-7.1

TABLE 7.1: PERFORMANCE OF NEOCOGNITRON ALGORITHM UNDER DIFFERENT CRITERION

Criteria	Neocognitron Algorithm
Number of Neurons	21301
Number of Unknowns	V.L.
Capacity	5
Effect of Noise in Weight (Random No. Added)	No effect
Effect of Increase of Weight	N.S.
Noise in Input	N.S.
Range of No. of Pixels that can made off	N.S.
Range of No. of Pixels that can made on	N.S.
No of Connection we can loose (wt=0)	N.S.

REFERENCES

[1] Basit Hussain and M.R.Kabuka, "A Novel Feature Recognition Neural Network and its Application to string Recognition",IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 16, No. 1,PP. 98-106, Jan. 1994.

[2] Z.B. Xu, Y.Leung and X.W.He," Asymmetrical Bidirectional Associative Memories", IEEE Transactions on systems, Man and cybernetics, Vol.24, PP.1558-1564, Oct.1994.

[3] Schalkoff R.J.," Pattern Recognition: Statistical Structured and Neural; Approach", John Wiely and Sons, 1992

[4] Fukuanga K., "Stastitical Pattern recognition-2nd ed.", Academic Press, 1990

[5] V.K.Govindan and A.P. Sivaprasad, "String Recognition- A review", Pattern recognition, Vol.23,No. 7, PP. 671-683, 1990

[6] Fu K.S., "Syntactic Pattern Recognition," Prentice Hall, 1996.

[7] Y.K. Chen, J.F. Wang, Segmentation of single or multiple touching handwritten numeral strings using background and foreground analysis, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1304–1317.

[8] Jacek M. Zurada, "Introduction to artificial Neural Systems", Jaico Publication House. New Delhi, INDIA

[9] C.L. Liu, K. Nakashima, H. Sako,H.Fujisawa,Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognition 36 (2003) 2271–2285.

[10] R. Plamondon, S.N. Srihari, On-line and o8-line handwritten recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 62–84.

[11] C.L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2004) 265–279.

DOI: 01.IJNS.02.03.200

[12] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic segmentation of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.
[13] U. Bhattacharya, T.K. Das, A. Datta, S.K. Parui, B.B.Chaudhuri, A hybrid scheme for handprinted numeral recognition basedon a self-organizing network andMLP classiHers, Int. J. Pattern Recognition Artif. Intell. 16 (2002) 845–864.
[14] C.L. Liu, H. Sako, H. Fujisawa, Effects of classifier structure and training regimes on integrated segmentation and recognition of handwritten numeral strings, IEEE Trans. Pattern Recognition Mach.Intell. 26 (11) (2004) 1395–1407.
[15] K.K. Kim, J.H. Kim, C.Y. Suen, Segmentation-based recognition of handwritten touching pairs of digits using structural features, Pattern Recognition Lett. 23 (2002) 13–24.
[16] Jain, A., Duin, P., Mao, J., 2000. Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Machine Intell. 22 (1), 4–37.
[17] Tax, D.M.J., van Breukelen, M., Duin, R.P.W., Kittler, J., 2000. Combining multiple classifiers by averaging or by multiplying?Pattern Recognition 33, 1475–1485.
[18] L.S. Oliveira, R. Sabourin , F. Bortolozzi , C.Y. Suen” Impacts of verification on a numeral string recognition system” Elsevier Science, Pattern Recognition Letters 24 (2003) 1023–1031.
[19] U. Pal, B.B. Chaudhuri “Indian script character recognition: a survey” Elsevier Science, Pattern Recognition, 37 (2004) 1887 – 1899
[20] Xiangyun Ye, Mohamed Cheriet ,Ching Y. Suen “StrCombo: combination of string recognizers” Elsevier Science, Pattern Recognition Letters 23 (2002) 381–394.
[21] Symon Haikin, “Neural Networks: A comprehensive Foundation”, Macmillan College Publishing Company, New York, 1994.
[22] Javad Sadri, ChingY. Suen, Tien D. Bui “Agenetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings” Elsevier Science, Pattern Recognition 40 (2007) 898 – 919
[23] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 2003.

DOI: 01.IJNS.02.03.200